

# ifSocket

## Implemented By

- `roDataGramSocket`
- `roStreamSocket`

## Supported Methods

- `Send(data as Object, startIndex as Integer, length as Integer) as Integer`
- `SendStr(data as String) as Integer`
- `Receive(data as Object, startIndex as Integer, length as Integer) as Integer`
- `ReceiveStr(length as Integer) as String`
- `Close() as Void`
- `SetAddress(sockAddr as Object) as Boolean`
- `GetAddress() as Object`
- `SetSendToAddress(sockAddr as Object) as Boolean`
- `GetSendToAddress() as Object`
- `GetReceivedFromAddress() as Object`
- `GetCountRcvBuf() as Integer`
- `GetCountSendBuf() as Integer`
- `Status() as Integer`

## Description of Methods

These are the basic binding and data transfer operations used on both `roStreamSocket` and `roDataGramSocket`. They are synchronous or asynchronous as determined by the socket's blocking behavior. If there is a valid assigned `roMessagePort`, the blocking behavior is considered asynchronous (non-blocking). Otherwise, the blocking behavior is considered synchronous.

### **Send(data as Object, startIndex as Integer, length as Integer) as Integer**

Sends up to length bytes of data to the socket. The data parameter is a `roByteArray` containing the data to be sent, starting at the byte indexed by `startIndex`.

Returns the number of bytes actually sent.

### **SendStr(data as String) as Integer**

Sends the whole string to the socket if possible.

Returns the number of bytes actually sent.

### **Receive(data as Object, startIndex as Integer, length as Integer) as Integer**

Reads up to length bytes from the socket. The data parameter is a `roByteArray` into which data is stored, beginning at the byte indexed by `startIndex`.

Returns the number of bytes actually read.

**ReceiveStr(length as Integer) as String**

Reads up to length bytes from the socket and stores the result in a string.

Returns the received string. If no bytes were received, the string will be empty.

**Close() as Void**

Performs an orderly close of socket.

After a close, most operations on the socket will return invalid.

On blocking sockets, this clears the receive buffer and blocks until the send buffer is emptied. Neither buffer may be read or written afterward.

On non-blocking sockets, both the send and the receive buffer may be read but not written.

**SetAddress(sockAddr as Object) as Boolean**

sockAddr is an roSocketAddress.

Returns true if successfully set address using a BSD bind() call.

**GetAddress() as Object**

Returns the roSocketAddress object bound to this socket.

**SetSendToAddress(sockAddr as Object) as Boolean**

Set remote address for next message to be sent

sockAddr is an roSocketAddress.

Returns true if successfully stored address as first half of underlying BSD sendto() call

**GetSendToAddress() as Object**

Returns roSocketAddress for remote address of next message to be sent

Can also be used to return the remote address on newly accepted sockets

**GetReceivedFromAddress() as Object**

Returns roSocketAddress for remote address of last message received via receive()

Can also be used to return the remote address on newly accepted sockets

**GetCountRcvBuf() as Integer**

Returns the number of bytes in the receive buffer

**GetCountSendBuf() as Integer**

Returns the number of bytes in the send buffer

**Status() as Integer**

Returns the errno of the last operation attempted or zero if the last operation was a success.